

NAME: _____

STUDENT #: _____

SIGNATURE: _____

CLASS & SECTION: _____

UNIVERSITY OF SASKATCHEWAN
Department of Computer Science

CMPT 111.3 FINAL EXAMINATION
December 11, 1995

Total Marks: 125

OPEN BOOK

Time: 3 Hours

This exam consists of short answer questions and programming problems. Read each question carefully and write your answer legibly **on this question sheet**. You may use the backs of pages or answer books for rough work but all final answers must be on the question sheet. The marks for each question are indicated in the margin. Allocate your time accordingly. Ensure that your **name AND student number** are clearly written on the examination paper and that your name is on every page. When you have finished the exam, hand in the examination paper and ALL answer books to your instructor.

Marks

5 1. True/False: Indicate TRUE or FALSE for each of the following:

- _____ a) The storage identified by a memory address is returned to free storage (deallocated) simply by setting its pointer to nil.
- _____ b) Binary trees are always ordered.
- _____ c) A queue can be implemented as a linked list.
- _____ d) A recursive procedure without a base case causes infinite looping.
- _____ e) An array is an example of a dynamic data structure.

5 2. Multiple Choice Questions: circle the letter that corresponds to the **one** best answer for each part.

a) What is the output of the following Pascal code fragment?

```
a := 'The';  
b := 'cat';  
c := 'sat';  
writeln(concat(a, copy(concat(b, c), 3, 2)))
```

- (a) Thecatsat
- (b) Thets
- (c) Theats
- (d) satcatThe

- b) A text file that is to provide information to a program should be initialized with the following procedure:
 - (a) rewrite
 - (b) append
 - (c) delete
 - (d) reset
- c) Queues operate in the following manner:
 - (a) last in, first out (LIFO)
 - (b) last in, last out (LILO)
 - (c) first in, last out (FILO)
 - (d) first out, last in (FOLI)
- d) Which of the following is not one of the four stages of the Waterfall model (Software Lifecycle)?
 - (a) Design
 - (b) Testing and maintenance
 - (c) Implementation
 - (d) Rapid prototyping
- e) Turbo Pascal always looks for variable references in the following locations:
 - (a) external, then global
 - (b) local, then global
 - (c) local, then external
 - (d) global, then local

16 3. a) Consider the following Pascal function named Foo.

```

function Foo (str: string): string;
var
    X: string;
begin
    if str = '' then
        Foo := ''
    else
        begin
            X := copy(str, 1, 1);
            delete(str, 1, 1);
            Foo := concat(Foo(str), X);
        end;
    end;

```

- i) What value will be written out after the following function call has been made? (5 marks)

```
writeln(Foo('Santa'))
```

ii) In twenty-five words or less, what does the function Foo do? (3 marks)

b) Consider the following type definition and function Boo.

Type

```
nodePtr = ^ node;  
node = record  
    item:integer;  
    link:nodePtr;  
end;
```

```
function Boo (head: nodePtr; X:integer): integer;  
begin  
    if head = nil then  
        Boo := 0  
    else if head^.item < X then  
        Boo := 1 + Boo(head^.link, X)  
    else  
        Boo := Boo(head^.link, X)  
    end;  
end;
```

i) Suppose that L is a pointer of type nodePtr that points at a linked list with nodes containing values: 3 6 7 2 5 8 2. What value will be written out after the following function call has been made? (5 marks)

```
writeln(Boo(L, 5))
```

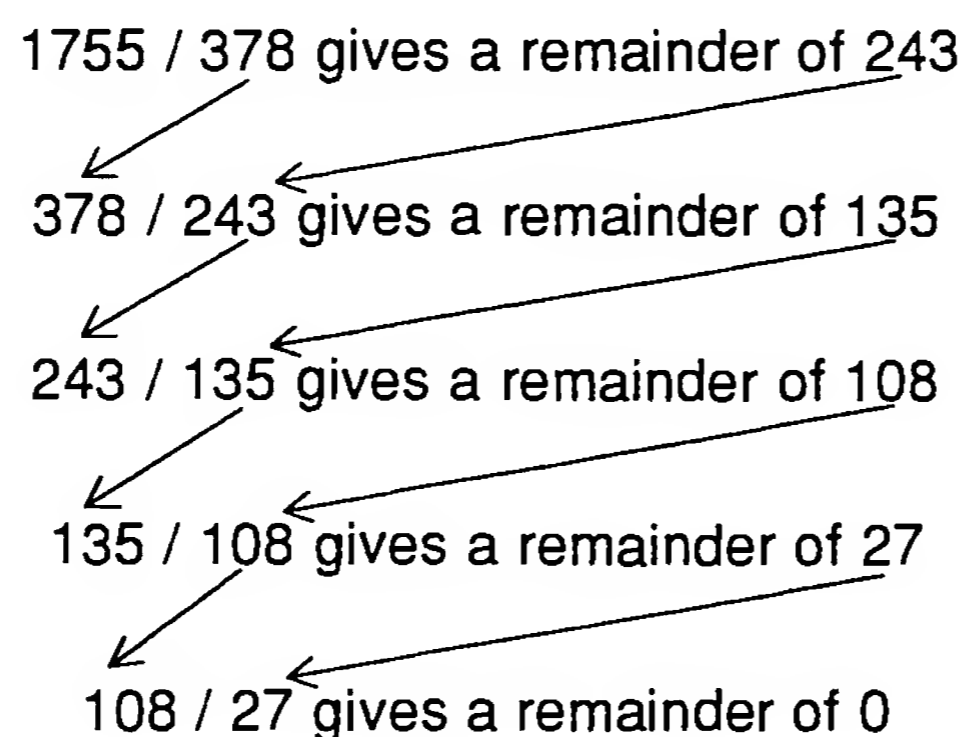
ii) In twenty-five words or less, what does the function Boo do? (3 marks)

10 4. Given an array of 31 elements containing the numbers 1 through 31 in ascending order:

- a) How many comparisons does it take to find the value 16
 - i) using a Linear Search?
 - ii) using a Binary Search?
- b) How many comparisons does it take to determine the value -1 is not in the array
 - i) using Linear Search?
 - ii) using Binary Search?
- c) In general, which search method is faster and why?
- d) Provide an example array in which Linear Search and Binary Search perform the same number of comparisons in searching for the same value.
- e) For an arbitrary input array is it possible for one search algorithm to find a value while the other does not? If so, provide an example.

14 5. **Euclid's algorithm** can be used to find the greatest common factor (GCF) of two integers. The GCF of two numbers is the largest number that divides evenly into both, e.g., the GCF of numbers 8 and 4 is 2. Euclid's algorithm involves repeatedly dividing and shifting two numbers until a remainder of 0 is attained. The last divisor resulting in a remainder of 0 is the GCF.

For example, given 2 integers 1755 and 378, Euclid's algorithm is applied as follows:

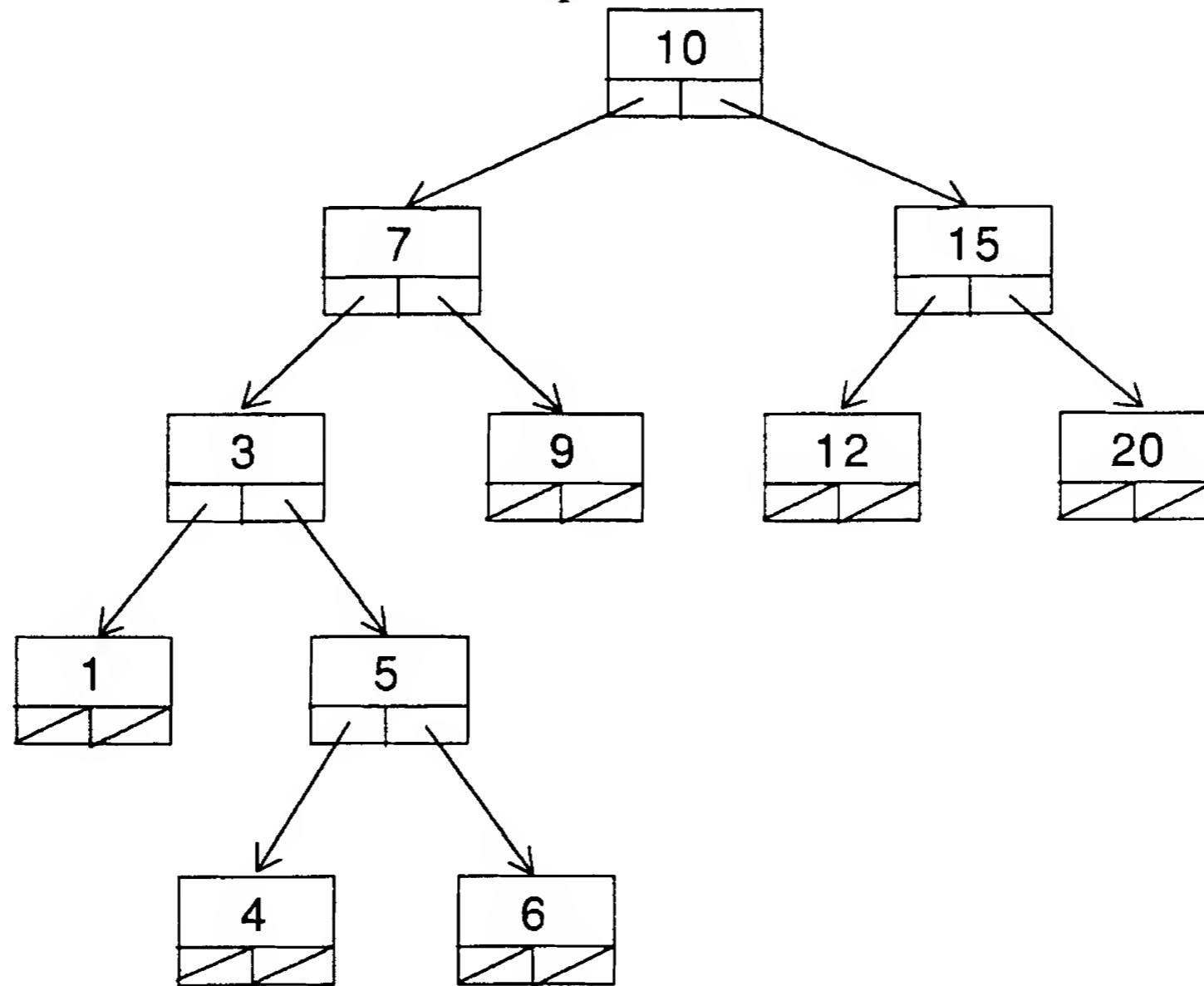


The last divisor, i.e., 27 is the greatest common factor of 1755 and 378.

Write a function that accepts 2 integers and returns the GCF of those numbers (an integer value). You can assume that the first integer is always larger than the second.

15 6. Given the following sorted binary tree, T,

*For the tree I would use 10 to 15 nodes with depth of 4 or 5.



- a) What is the output of a:
 - i) Preorder traversal?
 - ii) Inorder traversal?
 - iii) Postorder traversal?
- b) If the value XX is inserted, what is the value of its parent's node?
- c) Starting from the root node, how many node comparisons does it take to determine that the value XX is not in the tree?
- d) The following procedure was written by a computer science student to print out the values in a sorted binary tree in descending order:

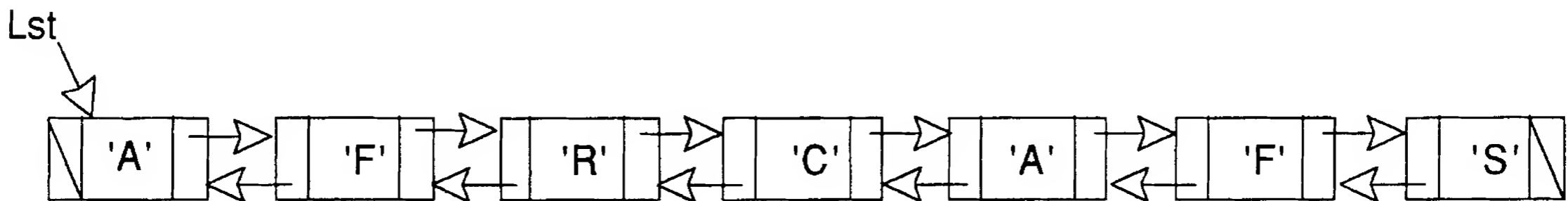
```

procedure Descend(T: tree);
begin
    if T <> NIL then begin
        Descend(T^.lptr);
        write(T^.info);
        Descend(T^.rptr);
    end;
end;

```

Unfortunately, this procedure does not work. Rewrite procedure Descend(T) so that the values are printed in descending order. For the above example tree Reverse(T) should output -----
-----.

- 20** 7. It is possible to create a variation of a linked list where each node points to the node immediately following itself, i.e., its successor, AND to the node immediately preceding itself, its predecessor. This specialized type of linked list, a doubly linked list, is useful for situations where there is a desire to traverse a list in two directions - forwards and backwards.



- a) Provide the necessary type definitions and var declarations to represent the linked data structure as illustrated above. (6 marks)
- b) Write a segment of Pascal code that searches list Lst for all occurrences of the character "F" and prints out the character in the node immediately preceding "F" and in the node immediately following "F". Assume that Lst has already been declared and contains zero or more nodes. Be sure to identify any additional variables that you may require and their types. Note, your approach should be general enough to handle any instance of a doubly linked list, not just the specific example above. (14 marks)

